**Sandro Costa**

**Dr. Tarek Ahmed**

**CMSC-203**

**04-05-2021**

**Assignment-05**

**Pseudocode:**

The class TwoDimRaggedArrayUtility will perform following actions with following methods:

Method readFile will read from a file and returns a two- dimensional ragged array (list of some values of same type) of double

If file is empty or contains nothing, will return a null or error message

Method writeToFile will writes the ragged array of doubles into the file. Each row is on a separate line within the file and each double is separated by a space.

If file is empty or contains nothing, will return a null or error message

Method getTotal will return the sum of all the elements in the two-dimensional array

If both arrays are empty or contains nothing, will return a null or error message

Method getAverage will return the average of all the elements in the two-dimensional array

If elements of both arrays are empty or contains nothing, will return a null or error message

Method getRowTotal returns the total of the selected row in the two-dimensional array index 0 refers to the first row.

Method ColumnTotal will return the total of the selected column

Method getHighestInRow will returns the largest element of the selected row

Method getHighestInRowIndex will return the index of the largest element of the row

Method getLowestInRow will returns the smallest element of the selected row

Method getLowestInRowIndex will return the index of the smallest element of the row

Method getHighestInColumn will returns the largest element of the selected column

Method getHighestInColumnIndex will return the index of the largest element of the column

Method getLowestInColumn will returns the smallest element of the selected column

Method getLowestInColumnIndex will return the index of the smallest element of the column

Method getHighestInArray will return the largest element in the two-dimensional array

Method getLowestInArrey will return the smallest element in the two-dimensional array

The HolidayBonus class will return:

The Bonus for each store in the district by calculateHolidayBonus Method

The total bonus will be calculated by calculateTotalHolidayBonus method

**UML diagram:**

|  |
| --- |
| TwoDimRaggedArrayUtility |
|  |
| +getAverage(data: double[][]): double  +getColunTotal(data: double[][], col: int): double  +getHighestInArray(data: double[][]):double  +getHighestInColumn(data: double[][], col: int): double  +getHighestInColumnIndex(data: double[][], col: int): int  +getHighestInRow(data: double[][], row: int): double  + getHighestInRowIndex(data: double[][], row: int): int  +getLowestInColumn(data: double[][], col: int): double  +getLowestInColumnIndex(data: double[][], col: int): int  +getLowestInRow(data: double[][], row: int): double  +getLowestRowIndex(data: double[][], row: int): int  +getRowTotal(data: double[][], row: int): double  +getTotal(data: double[][]): double  +readFile(file: File): double  +writeToFile(data: double[][], outputFile: File): void |

|  |
| --- |
| HolidayBonus |
|  |
| +calculateHolidayBonus(data: double[][], hight: double, low:double): double[]  ++calculateTotalHolidayBonus(data: double[][], hight: double, low: double, otherL double): double |

**Test Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Input** | **Actual Input** | **Expected Output** | **Actual Output** | **Did the test pass?** |
| 1 | {1 2 3}  {4 5}  {6 7 8 } | {1 2 3}  {4 5}  {6 7 8 } | $6.00 $9.00 $21.00 $11.00 $14.00 $11.00 $3000.00 $15000.00 $36.00 $22000.00 |  |  |
| 2 | {1.65 4.50 2.36 7.45 3.44 6.23}  { 2.22 -3.24 -1.66 -5.48 3.46},  { 4.23 2.29 5.29}  {2.76 3.76 4.29 5.48 3.43}  {3.38 3.65 3.76}  {2.46 3.34 2.38 8.26 5.34} | {1.65 4.50 2.36 7.45 3.44 6.23}  { 2.22 -3.24 -1.66 -5.48 3.46},  { 4.23 2.29 5.29}  {2.76 3.76 4.29 5.48 3.43}  {3.38 3.65 3.76}  {2.46 3.34 2.38 8.26 5.34} | $25.63 -$4.70 $11.81 $19.72 $10.79 $21.78 $85.03 $16.70 $14.30 $16.42  $15.71 $15.67 $6.23 $17,000.00 $4,000.00 $12,000.00 $9,000.00 $6,000.00 $16,000.00 $64,000.00 | $25.63 -$4.70 $11.81 $19.72 $10.79 $21.78 $85.03 $16.70 $14.30 $16.42  $15.71 $15.67 $6.23 $17,000.00 $4,000.00 $12,000.00 $9,000.00 $6,000.00 $16,000.00 $64,000.00 |  |
| 3 | { 1.0, 2.0, 3.0, 4.0 },  {5.0, 6.0 },  { 7.0, 8.0, 9.0 },  {10.0, 11.0},  {12.0, 13.0, 14.0} |  | $1.00 $2.00 $3.00 $4.00 $10.00 $8000.00  $5.00 $6.00 $11.00 $4000.00  $7.00 $8.00 $9.00 $24.00 $6000.00  $10.00 $11.00 $21.00 $4000.00  $12.00 $13.00 $14.00 $36.00 $15000.00  $35.00 $40.00 $26.00 $4.00 $105.00 $37000.00 |  |  |